IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application for:

DATAFLOW GRAPH COMPRESSION FOR POWER REDUCTION IN A VECTOR PROCESSOR

Inventors: Philip E. May
Brian Geoffrey Lucas
Kent Donald Moat

Docket Number: **CML00770D**

DATAFLOW GRAPH COMPRESSION FOR POWER REDUCTION IN A VECTOR PROCESSOR

CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to application serial number 10/184,772 (CML00108D) titled "Scheduler for Streaming Vector Processor", filed June 28, 2002 and to application serial number 10/184,583 (CML00107D), titled "Reconfigurable Streaming Vector Processor", filed June 28, 2002.

FIELD OF THE INVENTION

This invention relates generally to the field of microprocessors. More particularly, certain embodiments consistent with this invention relate to a method and apparatus for dataflow graph compression for power and memory reduction.

BACKGROUND OF THE INVENTION

VLIW (very long instruction word) processors are characterized by their ability to process multiple instructions in parallel using different functional units within the processor. Other processors, such as dual-path processors have a similar capability. A VLIW instruction comprises a number of sub-words or fields, each of which controls a functional unit within the processor. Fully parallel operation, in which all functional units are used at each time step, is rare. More commonly, many

instructions contain "NOP" instructions in several of the sub-words or fields, indicating that the corresponding functional unit is not operated at that time step.

Power reduction in processors controlled by VLIWs or other multiple-instruction words is very important in handheld computing devices, such as PDAs, digital cameras, navigation devices and cellular telephone handsets that rely on battery power. Power reduction can be used to reduce the weight and/or increase the operating time of a device. Two techniques for power saving are (i) reducing memory accesses by reducing the size of the program of instructions (compression), and (ii) disabling hardware resources (memory and functional units) when they are not required.

VLIW compression schemes are used in many VLIW architectures (e.g., IA-64, StarCore, TI DSPs). The goal of these schemes is the elimination of all NOPs from the instruction stream. This reduces the memory requirements for the code, and it reduces the memory bandwidth required to fetch instructions.

In one method of power saving, special instructions are added to a processor to shut down the datapath elements under program control. In a further method, datapath elements are disabled based on an instruction pre-decode stage. The instruction words are examined and the datapath is dynamically disabled on an instruction-by-instruction basis. A disadvantage of these approaches is that they add to dataflow graph complexity.

In one method of compression, NOPs are eliminated from the code by rearranging the slices within VLIW words, so that NOP fields in sequential VLIWs line up (are at the same location in the VLIW word). This allows banks of memory to

be powered off for periods of time, saving power. A disadvantage of this method is the complexity associated with rearranging the slices within VLIW words.

OVERVIEW OF CERTAIN EMBODIMENTS OF THE INVENTION

Certain embodiments consistent with the present invention relate generally to memory and power saving in microprocessors that use multiple-instruction control words. Objects, advantages and features of the invention will become apparent to those of ordinary skill in the art upon consideration of the following detailed description of the invention.

In accordance with certain embodiments of the invention is a mechanism for efficiently eliminating NOPs from multiple-instruction control words, while at the same time disabling unused elements in the processor and unused instruction memory banks. A multiple-instruction control word comprises a number of ordered fields, with each field containing an instruction for an element of the processor. The sequence of instructions for a loop is compressed by identifying a set of 'aligned' fields that contain NOP instructions in all of the control words of the sequence. The NOP instructions are 'aligned' if they occur at the same position within the control word, i.e. they are associated with the same element of the processor. The sequence of control words is then modified by removing the fields in the identified set of aligned fields containing NOP instructions and adding an identifier that identifies the set of fields removed. The sequence of control words is processed by fetching the identifier at the start the loop, then, for each control word in the sequence, fetching a

control word and reconstructing the corresponding uncompressed control word by inserting NOP instructions into the compressed control word as indicated by the identifier. The identifier may be a bit-mask and may used to disable memory units and processing elements for the duration of the loop to reduce power consumption by the processor.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself, however, together with objects and advantages thereof, will best be understood by reference to the following detailed description of certain illustrative embodiments consistent with the present invention when read in conjunction with the accompanying drawing(s), wherein:

FIG. 1 is an exemplary sequence of multiple-instruction control words.

FIG. 2 is an exemplary sequence of compressed multiple-instruction control words in accordance with certain aspects of the invention.

FIG. 3 is an exemplary compression mask.

FIG. 4 is a diagrammatic representation of a system in accordance with certain aspects of the present invention.

FIG. 5 is a histogram illustrating processor power reduction using the present invention.

DETAILED DESCRIPTION

While this invention is susceptible of embodiment in many different forms, there is shown in the drawings and will herein be described in detail one or more specific embodiments, with the understanding that the present disclosure is to be considered as exemplary of the principles of the invention and not intended to limit the invention to the specific embodiments shown and described. In the description below, like reference numerals are used to describe the same, similar or corresponding parts in the several Views of the drawings.

The invention is described with reference to a re-configurable streaming vector processor (RSVP). However, the invention is applicable to other processors that use multiple-instruction control words.

The RSVP is a statically scheduled VLIW machine that executes dataflow graphs on vector data (data streams) in a highly pipelined fashion. Typically, a dataflow graph is a representation of the entire inner loop of a computation. The VLIW nature of RSVP can be exploited to reduce the power in the RSVP datapath by disabling the portions of the datapath that are not utilized by a particular dataflow graph. This mechanism is simple, inexpensive, and provides significant power savings.

A dataflow graph in RSVP is represented as a collection of VLIW control words that are executed repeatedly as a loop. These control words contain sub-words or fields, each of which is responsible for controlling a portion of the datapath (datapath element). An exemplary dataflow graph is shown in **FIG. 1**. Referring to

FIG. 1, the top four rows of the table represent VLIW control instructions. In this example, the processor has ten datapath elements. Each VLIW comprises ten ordered fields, one for each of the datapath elements. Thus, the columns of the table contain the instructions for each datapath element. Each column of the table is also described herein as a set of aligned fields. The asterisks denote instructions other then NOP instructions (No Operation). Thus, for example, VLIW 1 contains NOPs for datapath elements 1, 3, 4, 6, 7 and 8, but does contain instructions for datapath elements 2, 5, 9 and 10. According to one aspect of the method of the present invention, all control fields that contain NOPs for all VLIWs in the loop are identified. If a column contains only NOP instructions, then the associated datapath element is not used in the execution of the graph.

According to one aspect of the present invention, any columns that contain only NOPs are eliminated when the executable code is created. In addition, the VLIWs are compressed by removing these NOPs. A compression mask, shown as the last row in the **FIG. 1**, is created to identify which columns have been removed. In this example, the compression mask has '0' values corresponding to the eliminated columns and '1' values for the remaining columns. Since the ordering of the fields is not affected by the compression, the original VLIWs may be reconstructed from the compressed instructions and the compression mask. In operation, the compression mask is passed to the processor and used to reconstruct the VLIWs. In one embodiment the compression mask is placed in a VLIW header word (part of the RSVP executable containing graph information and characteristics) and may be used in VLIW reconstruction and datapath element and memory bank disabling.

This approach is in contrast to prior techniques since the entire data graph is considered in the compression operation and only the NOPs that are naturally aligned are removed. The fields in the VLIWs are not rearranged to line up NOPs. Thus, the compression technique of the present invention is simple to implement and may be automated.

The resulting compressed VLIWs are shown in **FIG. 2**. In **FIG. 2**, there are no instructions for the datapath elements 3, 4 and 7, since these elements are not used in the execution of the graph. The corresponding compression mask is shown in **FIG. 3**. Indicators other than a bit-mask may be used without departing from the present invention.

One embodiment of a system for processing VLIW instructions is illustrated in **FIG. 4**. Referring to **FIG. 4**, when a graph is executed, the header is fetched first, and the compression mask is latched in the mask latch 102. The compression mask is presented to logic device 104 that decodes the mask and enables the correct number of memory banks 106 in the VLIW memory as well as programming the pipelined permute unit 108. As VLIWs are transferred from main memory to the VLIW memory 106, only the enabled banks are used, and when the graph executes (from VLIW memory), the disabled banks draw minimal power. The pipelined permute unit 108 uses the compression mask to reconstruct the VLIWs by placing NOP operation codes in the fields indicated by '0' in the mask. These complete VLIW words are placed in VLIW register 110 and presented to the datapath for execution.

In addition, the compression mask is passed via link 112 to the datapath elements and is used to enable or disable the datapath elements. Since the mask

locations map 1:1 to the datapath elements (control fields), the raw mask bits may be used to clock gate the datapath. The datapath element shutdown is initiated using the compression mask supplied in a header word associated with a dataflow graph. This mask is examined once, and for the duration of the dataflow graph in question, the affected function units remain disabled.

The task of MPEG4 encoding on an RSVP is now considered as an example. In MPEG4 encoding on the RSVP, there are 18 separate dataflow graphs. Upon averaging all 18 graphs, 52% of the datapath control fields in the VLIW word are unused, and the corresponding datapath elements can be disabled for the duration of the graph. In addition, approximately half of the VLIW memory banks can be disabled.

FIG. 5 shows a breakdown of these graphs as a histogram, with each bin corresponding to 10% VLIW field disabling. Each bar indicates the number of graphs that fall in that percentage. The largest single bin is 60% of VLIW fields (datapath elements) disabled, and over half of the graphs (11) have 60% or greater of their datapath elements disabled.

The present invention combines both memory and datapath power reduction in a single mechanism.

Those of ordinary skill in the art will recognize that the present invention has been described in terms of exemplary embodiments based upon use of dataflow graphs for an RSVP. However, the invention should not be so limited, since the present invention could be implemented using hardware component equivalents such as special purpose hardware and/or dedicated processors, which are equivalents to the

invention as, described and claimed. Similarly, general purpose computers, microprocessor based computers, digital signal processors, microcontrollers, dedicated processors, custom circuits, ASICS and/or dedicated hard wired logic may be used to construct alternative equivalent embodiments of the present invention.

While the invention has been described in conjunction with specific embodiments, it is evident that many alternatives, modifications, permutations and variations will become apparent to those of ordinary skill in the art in light of the foregoing description. Accordingly, it is intended that the present invention embrace all such alternatives, modifications and variations as fall within the scope of the appended claims.

What is claimed is: